ALL ABOUT
AMOS

VOLUME 1   **THE ALTERNATIVE AMOS RESOURCE**   **ISSUE 1**
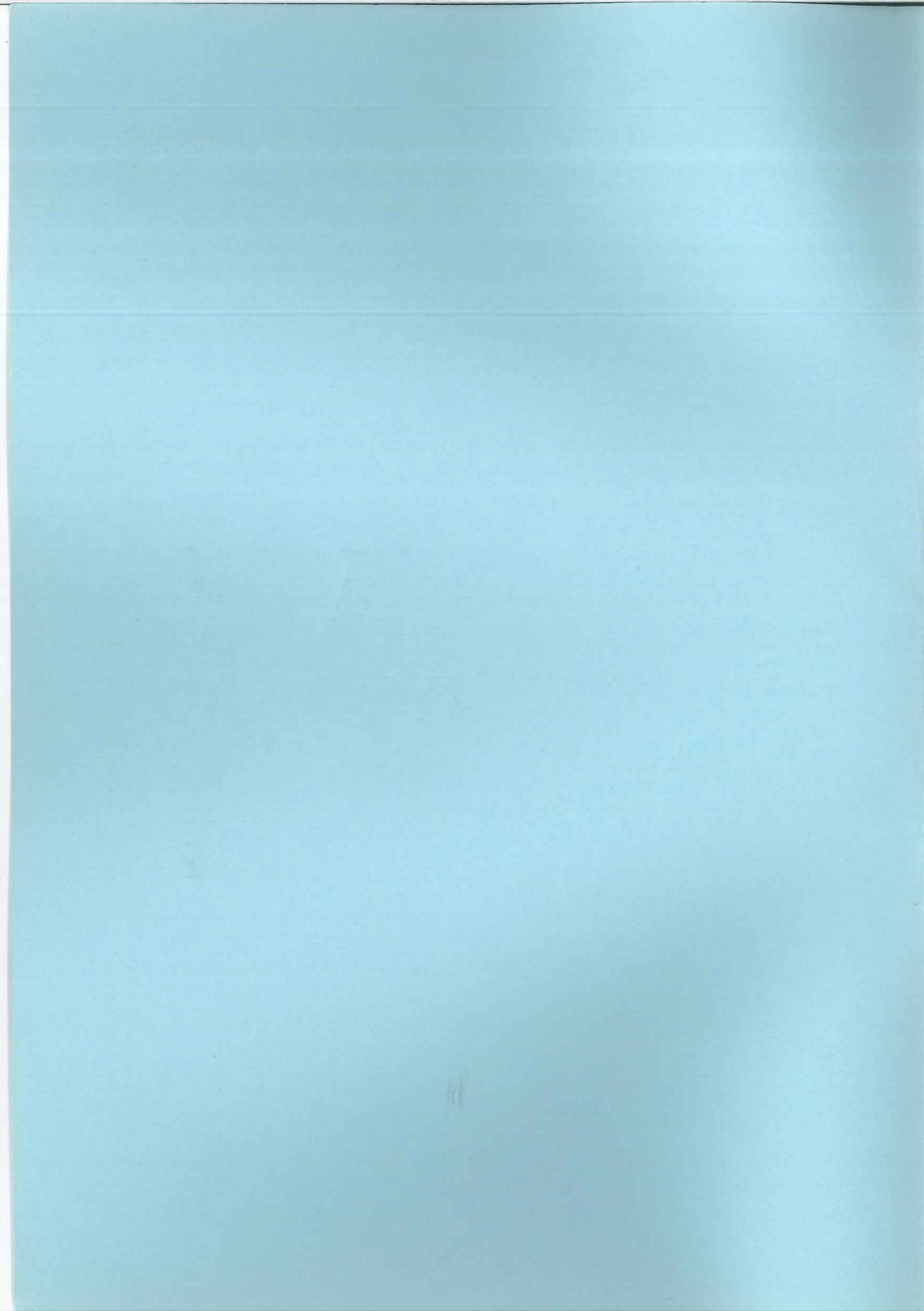
1ST ISSUE CONTENTS

REGULARS

PREVIEWS & REVIEWS

FEATURES

CREDITS

The contributors and friends
who deserve the credit for
this brilliant first issue are

Colin White
Gary Symons
Richard Vanner
Anne Tucker
Leo Douglas
Colin The Clown
Chris Payne

EDITOR STRIKES BACK!!

WELCOME!!!

Welcome to the very first issue of "ALL ABOUT AMOS". Before all of the good bits I would like to explain exactly why "ALL ABOUT AMOS" has come into existence. We are at a very exciting point in the development of AMOS as a computer language; there are now two commercial extensions which add new commands to AMOS (TOME and AMOS 3D) as well as the Licenseware CTEXT extension - which is available from Déjà Vu Software. Fun School 4 is now finished (90% of which is written in AMOS) and of course if you have not heard the news yet, EASY AMOS should be available around January time. I feel that although AMOS is supported by a few off-the-shelf magazines people still need to know more about programming and making the most of this brilliant language.

This first issue is rather special, it is mainly written by us, for you. In the future if you want to see something featured in "ALL ABOUT AMOS" write in and tell us, after all you are paying a subscription so make the most of your money by influencing the contents. In every issue we will offer a prize for the best article written by a reader, not just any old prize but something really useful like a RAM upgrade, external drive, an audio digitizer or a video digitizer (actually a list will be sent to the winner of all the fabulous prizes you can choose from). As well as that, the prize winning article will be printed in these very pages (of course if no one sends anything interesting then no one gets the prize!). What do you think? Is it a good idea? Fill in the readers survey, this is your chance to shape the future of YOUR magazine.

HELP US GROW

If you enjoy this first issue why not get your AMOS owning friends to subscribe? We reckon that for every two hundred people that subscribe to A.A.A. we can add a page to each issue. That's five pages for every thousand, and considering there are about 40,000 AMOS owners out there we could end up with a very BIG mag one day!

Peter Hickman.

PD, OR NOT PD?

As I am sure you already have gathered this is a magazine that will try to cater for all of the different types of AMOS users out there, from beginners to experts. For this reason I have written this little article which tells you all about so-called Public Domain software. With many different schemes run by PD libraries I thought it would be a good idea to give you a short explanation of the most important ones.

WHAT IS PD?

PD (Public Domain) software is written by people who want to share it with others. They have not written it for financial profit or gain, but for the sheer joy of programming. Because the authors are not motivated by money all PD software is free.

WHY CHARGE FOR FREE SOFTWARE?

Although PD is strictly free, the people who run the libraries which supply the software charge you for a number of things. You basically pay for the disk, the postage, miscellaneous expenses such as electricity & machine maintenance and finally something that a lot of people forget is that you are paying for a service. It is quite an expensive job running a big PD library, many of the good ones (such as Sandra Sharkey's Déjà Vu Software) offer regular postal updates and special money saving offers for regular customers. Of course once you have your copy of the latest PD version of "Teenage Spider Killers From the Planet Mango" you can give copies to your friends and spread it around to your hearts content, remember you can only do this legally with software which is Public Domain.

WHAT ABOUT SHAREWARE?

Shareware is a scheme whereby you purchase your software from the Public Domain library as usual. If you decide the software is good and you intend to use it more than a couple of times, a small fee (which is entirely voluntary) should then be sent to the author. In return for this fee you will usually receive not only the authors thanks for feeding his/her starving children but also a copy of the latest version or some other registration freebie. Probably the most well known piece of Shareware available in the U.K. is LLAMATRON by Jeff Minter.

WHAT IS LICENSEWARE?

The trouble with Shareware payment is that many people are lazy, dishonest, or just plain forgetful (that includes me!). This means that the authors often get very little reward for their hard work and never release any more software. Licenseware is a bit like a cross between budget commercial software and Shareware, with the PD Library distributing the product being LICENSED by the author of the program to sell his/her product - FOR A FEE! The Library charges slightly more than strict PD, then pays a small royalty to the program author for each copy sold. The copyright for each piece of Licenseware is still held by the author, and any unauthorised copying

or selling of the program is a criminal offence, so when buying any sort of Licenseware it is always worth dealing with a reputable company.

WHERE CAN YOU GET IT?

The company with the best selection of AMOS Public Domain, Shareware and Licenseware I have come across is Déjà Vu Software. This operation, run by the amazing Sandra Sharkey, started life as the OFFICIAL AMOS PD Library so they have the approval of Europress Software. Most of the software reviewed in this magazine will be available from this library. You can contact Déjà Vu Software at

25 PARK ROAD
WIGAN
WN6 7AA
ENGLAND

☎ 0942 495 261.

Remember to tell whoever you speak to where you heard about the Library!

RELIEF TIME

Well, with John Major's suits and the release of Workbench 2 on the A500+ the whole world seems to be going grey crazy. Now I am not saying that I find grey boring but I have purposely changed my workbench colours to purple and green, no one is going to say I lead a sad grey life!!! The really unfortunate thing is that people are actually paying me to write software with grey boxes galore, so I thought that a few procedures to do those horrid things would go down well. Do you really need instructions for these procedures? No I didn't think so, just type 'em in and drift off to sleep.....

GREY BOXES LISTING

```
Screen Open 1,640,200,4,Hires
Flash Off
Curs Off
Cls 0
Palette$CCC,$FFF,$AAA,$333
OUT_BOX[80,10,200,50]
IN_BOX[360,10,200,50]
End

Procedure IN_BOX[X,Y,XL,YL]
Ink 2
Bar X,Y To X+XL,Y+YL
Ink 3
Draw X,Y To X+XL-1,Y
Draw X,Y To X,Y+YL
Draw X+1,Y To X+1,Y+YL-1
Ink 1
Draw X+1,Y+YL To X+XL,Y+YL
Draw X+XL,Y To X+XL,Y+YL
Draw X+XL-1,Y+1 To X+XL-1,Y+YL
End Proc

Procedure OUT_BOX[X,Y,XL,YL]
Ink 2
Bar X,Y To X+XL,Y+YL
Ink 1
Draw X,Y To X+XL-1,Y
Draw X,Y To X,Y+YL
Draw X+1,Y To X+1,Y+YL-1
Ink 3
Draw X+1,Y+YL To X+XL,Y+YL
Draw X+XL,Y To X+XL,Y+YL
Draw X+XL-1,Y+1 To X+XL-1,Y+YL
End Proc
```

FAME

If you have a program or article that you would like to share with other readers of "ALL ABOUT AMOS" why not send it in? The best reader contribution published in every issue will receive a smashing prize. You can write about or program anything, the only limitation being that we cannot really include Sprites or Music.

IT's EASY AS ABC

By Richard Vanner

It's not often I get a chance to write for magazines or newsletters, simply because the job of software management keeps me extremely busy. But when Peter asked me to preview our new product Easy AMOS I just couldn't resist.

WHAT IS EASY AMOS?

Well, there are a number of differences with the original version of AMOS, but essentially it is an AMOS that has a cut-down command list, has many simple to use accessories and is accompanied by a detailed, easy to read manual.

Now all (or most) of you out there will have purchased the original AMOS and may be thinking "why would I want this cut-down version?". Just read on, you'll be delighted by what it has to offer.

THE SOFTWARE COMPONENTS

In Easy AMOS you get the following:

● Easy AMOS

This is the main Easy AMOS editor. From here you write and run your programs. There are some extra benefits here. The editor's menu (where Run, Test etc. are) and the file selector are now designed with more colours and prettier graphics. But the most beneficial addition is the Easy AMOS Tutor.

The Tutor is a part of Easy AMOS that allows you to single step through your programs, very much like a machine code monitor. For example, imagine

you have written a space invaders game. First you would load it into the editor, then select the Tutor option from the menu. Once inside the Tutor part of Easy AMOS, the program you had been editing is automatically ready for checking. Using the many icons on the Tutor's screen, you can step through the program and keep track of any variables or expressions. In the top left of the screen is a quarter size representation of the screen being created by your program. The fun and fascinating part of this is that you can let your program run in fast mode under the tutor while watching it execute in this small window.

So the Tutor as you can see is an invaluable feature of Easy AMOS for chasing out annoying bugs.

● Bob Editor

In Easy AMOS there are no sprite commands. This is to make the introduction to programming easier to grasp. The last thing we want to do is boggle the minds of new users with complex hardware sprite usage. We've also removed AMAL, powerful as it is, it just goes beyond the first year course of learning how to write programs!

We know a large number of AMOS users didn't like the original AMOS Sprite/Bob editor. It was written by Aaron Fothergill and not François Lionet (the author of AMOS). It was hard for Aaron to write -AMOS wasn't fully complete and difficult for Francois to help Aaron with it. So knowing about its weaknesses from your registration cards, we decided a new and easier editor was required. I won't

go into detail, just trust me, it's wicked - what do you think of it Ed?

● AMOS Disc

Ever wanted to format a disc or copy some files while still in AMOS?. Well AMOS Disc solves all your problems. This is a Disc Master or SID type program written in Easy AMOS. It stops the user having to multi-task back to WorkBench, which, on a 1 drive system can be a nightmare. The best thing about this program is the EXAMINE command. You select files and folders that need to be 'examined' and then click on the EXAMINE icon, each file found will be interrogated by AMOS Disc, if it's a sprite file you can choose to see the sprites, if it's a screen you can display it, if a music bank file is found you can play it and so on. All I need to say is that François and I have always used Disc Master, we've now pinched all its ideas and improved many areas and won't be going back to it!

● Games

We wanted to show that Easy AMOS could be used to write decent arcade games. We also wanted to explain how these games were written.

Tricycle Race (currently under development by François) is a super sprint type game but around a house! Can't tell you much yet as I've not seen a byte of it - knowing François it'll be amazing.

Block Buster is a breakout game.

ALL ABOUT THE
AUTHOR

Richard Vanner is the glue that holds Europress software together. Before becoming a Project Manager he was an expert programmer and even dabbled a little in Journalism, some of his work having appeared in the now defunct ATARI USER magazine.

Structured in such a way that all users can dissect the game to see how the fundamental elements of the game are put together. This one was written by Ronnie Simpson who wrote the amazing "Arcadia" game in AMOS (available from Déjà Vu Software now! Ed.).

● Easy Database

We also wanted to show that Easy AMOS isn't just a games maker! A label AMOS has had trouble shrugging off. So this card index program demonstrates AMOS's potential and shows the user how to control random access files.

● Challenge

Imagine you've just read the section on Bobs. Did it embed itself among the other trivia clogged up in your brain? The best way to check is to run the Easy AMOS Challenge game. Here you select a subject, in this case Bobs, and then have to answer 20 trivia questions about Bobs. The program reinforces what you have learnt. When you have over 80% you pass a subject, when you have completed all the subjects you receive a visual Easy AMOS diploma.

THE MANUAL

This is the most important part of Easy AMOS. To give users an idiots guide to such a huge subject is quite a big problem! This is the reason the AMOS

manual has a rather cold, no nonsense style. In stark contrast Easy AMOS has a comforting and humorous style, taking you step-by-step through the programming learning curve. Written by Mel Croucher (he of early 80's Pi-Man fame), the manual is fun to read and contains many silly cartoons of the AMOS character.

All software is covered in the manual, from the Bob Editor to the technicalities of screen swapping. A most enjoyable read and one that will help AMOS users who find their manual a little too deep.

RELEASE FORMATS

As we have yet to decide on these, I'll just say that Easy AMOS will be sold as a single package like its big brother AMOS. But it will also be available at a reduced rate to existing AMOS users.

AMOS V2.0

After Easy AMOS is complete we intend on upgrading AMOS further. All the new elements of Easy AMOS will be incorporated.

Ed's Comments

Well, is Easy AMOS the answer to your programming nightmares? The software is proven, the manual author is well known as a producer of quality (not to mention very humorous) writing and the accessories promise amazing things.

I have seen the package in it's current state and to tell you the truth it does look good, I only wish that a programming language like Easy AMOS existed when I started out on the ZX81!!!

DOS REPLACEMENT?

The lack of DOS/SHELL type commands in AMOS has always been frustrating for me. Up until a few months ago I was using a vanilla A500 with two drives, this makes the Amiga's wonderful multi-tasking capability pretty useless. So with no practical way to access the Amiga-DOS SHELL I came up with a few procedures to emulate some of the more useful functions.

This first one is a file copy routine which takes advantage of the AMOS banks. If you are a beginner you can think of "banks" as temporary storage areas inside your Amiga, these storage areas can be easily defined and manipulated within AMOS. The procedure FCOPY simple looks for the file passed into S$, finds out its length and then asks the Amiga to set aside a bank of memory large enough to hold a copy of the program. This area of memory is then saved back to disk under the name held in the variable D$.

Has anyone out there produced a routine to format an AmigaDOS disk? If you have any interesting DOS work-a-like routines why not send them and let other AMOS users benefit from your experience.

```
Procedure FCOPY[S$,D$]
Open In 1,SOURCE$
A=Lof(1)
Close 1
Reserve As Work 15,A
Bload SOURCE$,Start(15)
FIN=Start(15)+Length(15)
Bsave DEST$,Start(15) To FIN
End Proc
```

7

CHRISTMAS CRACKER!!

BY COLIN WHITE

'Twas the night before Christmas and all through the house, not a creature was stirring, not even a mouse. But wait... what's this? It's a weary ALL ABOUT AMOS columnist hunched over a hot Amiga. What's he up to? Reviewing a Christmas offering from Déjà Vu Software, that's what.....

This is the second in the SPARX "Just For Fun!" series and is basically a compendium of Christmassy bits 'n' pieces.

When the disk is first booted, the user is asked to enter the personal details of the child that the disk is for, so obviously if you've bought it for your kiddy, boot it up, enter the child's information, write enable the disk and the program does the rest. Next time the disk is booted - voila! - a personalised label is displayed with all the details you entered the first time. The package consists of three individual programs, each offering a treat based around the Christmas theme.

CHRISTMAJIG

A Christmas version of "Thingamajig"/"Colouring Book" with six festive black and white pics for the child to colour, and a "Puzzle" option which cuts up the picture and then prompts the user to replace the pieces in the correct places.

SANTA'S SLEIGH RIDE

A brilliant scrolling Santa game. You're Mr S. Claus, world famous bearded guy and reindeer keeper, and you've got to deliver all those lovely goodies to the eager kiddies by flying over the rooftops and dropping presents down the chimneys. On the upper levels of the game, obstacles appear to hinder you. Robins perch on chimney pots blocking them off, and mischievous elves drop down from the top of the screen, reversing your joystick controls if you hit them, but fear not - ol' St. Nick has a handy supply of snowballs and a mean throwing arm capable of knocking those birdies off the roofs. As for the elves - just avoid 'em!

STAR PROGRAM!

CHRISTMAS PAIRS

Hmmm....this sounds familiar.... Yes, you've guessed it! This is a nuclear reactor simulation! No - seriously - it's (surprise, surprise) a matching pairs card game. But it is better than your run-of-the-mill pairs games. Firstly it's got very pretty festive pictures. Secondly, the game responds with a manic "Ahaa!" cry when correctly matching a pair (very strange). Thirdly, eerrrmm....that's it really, but it is very good.

Credits can be obtained by clicking on the star on the Christmas tree (if you're really a scrolling text freak - Yaawwnnnn......Zzzzzzzz......etc.)

The personalised label on the opening screen is a fantastic idea for the kiddies and is bound to give them the feeling of owning the disk - congrats to Sparx for that idea, but it's implementation does have one down side. If you don't enter the personal details at the start, write enable the disk and allow it to save the label data, you will be faced with the infuriating prospect of being prompted to enter the details every time you quit from one of the programs or reboot the disk. The opening picture-menu(?) contains a very homely looking fire with excellent flickering flame animations, over which a selection of coloured stockings hang (Rather dangerously in fact. Quick, pass the fire extinguisher, granny!). Another feature worth mentioning is the screen fading, which is a lot smoother than in some previous Sparx offerings.

The Stocking Filler really gives a cute 'n' cosy feel of Christmas especially with those extra special seasonal touches. ("Ahhhh, where's my coffee and Bailey's, open fire and 'Johnny Mathis Sings Christmas' album?" Ed.). With all the Christmassy tunes an' jingly sleigh bells wandering all over the background, the festive feel oozes from every byte, chip 'n' pixel. A must for young Amiga users' Christmas stockings indeed! If we gave ratings, this program would get 9 out of 10 (it would be a ten but for that bloomin' label input routine.....).

AN ALTERNATIVE VIEW FROM THE EDITOR

As you will have seen on the opposite page this program wins the "ALL ABOUT AMOS" STAR PROGRAM award, which in basic terms means that it's the best program that we have reviewed this month!

I wish more people would produce seasonal software like this 'cos it's really fun, not just for the kids but for all of the family.

As Colin will testify, I love to get into the Christmas spirit at around the end of October. I play lots of seasonal records by Bing Crosby and other mouldy old crooners (all borrowed from my parents collection of course!) and start to pick up little Christmassy trinkets when I have to go shopping. STOCKING FILLERS is going to offer me another way to torture my family for a little longer.....

Remember folks, Christmas is a time for giving, so why not pick up the phone and order a copy from Déjà Vu Software now. At £3.50 it really would make an ideal stocking filler.
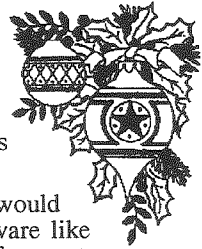
How often do you have to check the position of a particular on-screen box or part of a picture? Well I have to do it a lot in order to match flashing borders with menu selections and for this reason I wrote a simple program which allows you to get the coordinates of the mouse when moving it around a loaded IFF picture. The coordinates are displayed in a bar across the screen, this can be moved up or down with the cursor keys (beginners- the cursor keys are the small group located to the right of your keyboard with arrows drawn on them, do not mix them up with the numeric keypad which has numbers as well as arrows). You can also change the mouse cursor by pressing key "1" for an arrow and "2" for a crosshair. Press the right mouse button to load a new picture.

While typing in this listing remember that when you spot a ➡ symbol the next printed line actually joins onto the current one, we just couldn't fit the whole thing across the page!!

```
Screen Open 7,640,16,2,Hires
Curs Off
Cls 0
Palette $0,$FFF
Change Mouse 2
Y=250
Screen Display 7,,Y,,9
Repeat
FILE$=Fsel$("","",     ➡
"Pick  a picture")
If FILE$<>""
Load Iff FILE$,1
X=Screen Width
Y=Screen Height
Wait Vbl
```

Coordinate Finder listing continued

```
Limit Mouse X Hard(0),     ➡
Y Hard(0) To X Hard(X),Y
Hard(Y)
Screen To Front 7
Repeat
A$=Inkey$
S=Scancode
If A$="1"
Change Mouse 1
End If
If A$="2"
Change Mouse 2
End If
If S=76
Add Y,-5
Screen Display 7,,Y,,
End If
If S=77 Add Y,5
Screen Display 7,,Y,,
End If
Screen 7
Print At(7,0);"Xposition:";X     ➡
Screen(1,X Mouse);" "
Print At(60,0);"Yposition:";Y     ➡
Screen(1,Y Mouse);" "
Screen 1
Until Mouse Key<>0
End If
Until FILE$=""
```

IFF

IFF is an acronym for Interchange File Format. It was developed by Commodore and Electronic Arts to allow different programs to share data without all of the hassles experienced when people have to cope with conversion routines. The most common IFF file is an ILBM (InterLeaved BitMap) picture. This is the format which Deluxe Paint and AMOS use to save their pictures.

AMOS ASSEMBLY

BY GARY SYMONS

Welcome to this first instalment of our regular Assembly Language programming column. In this issue I will be giving an overview of Assembler and Machine Code (Machine Code being the product of Assembler); hopefully dispelling some of the myths and slander attributed to these programming languages.

No doubt you've been told writing in Assembler is painful and takes many years to master. Firstly Assembler is as painful as you program, so if you're a bad programmer in BASIC you'll be bad in Assembler. Secondly you can become an Assembler master within a year, but you must be prepared to sit down and try out many programs in order to acquire a good knowledge of the computer you are using (which in our case is the Amiga). Once you master your first computer and its associated Assembly Language you will master others within weeks as all computers are similar.

Before I begin I would like to say that a good knowledge of programming in any high level language would be VERY advantageous. This is because programming is the same in essence at high level and low level; so if you're hot at programming games etc. in AMOS there is no real reason why you can't be the next adonis gracing the centre pages of a games magazine.

So what is Assembler I hear some people shout??!!

Well before I answer that lets talk about Machine Code. Basically numbers represent operations,hence the term CODE as the MACHINE decodes these numbers into operations. To understand how Machine Code runs we must consider memory.

Your computer has memory and memory is divided up into locations each of which can store a byte (a number 0 to 255). We can make the computer put a number from 0 to 255 into any location (write) or we can make the computer tell us what number is in a certain location (read). To know which location we are ordering the computer to read or write, each location has an address. For example a postman uses an address to find the house to deliver a letter to (similar to the computer writing to a location). A courier has to know the address of your house to pick up a parcel (similar to the computer reading a memory location).

As you can see memory is useful for storing data. Even though it can only store numbers these numbers can represent letters, it can also represent many other things such as graphics and music which will be talked about in future issues of "ALL ABOUT AMOS".

Memory is also used to hold Machine Code as these are just numbers. The computer starts at a certain location, reads the number and decodes it into an operation (in this case the numbers correspond with tasks that the computer will carry out). The computer then advances to the next location and repeats the process. On many of the new computers including the AMIGA the computer reads (or fetches) from more then one location at a time. This
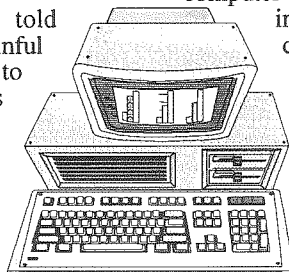
means larger numbers which in turn means a larger number of more sophisticated operations.

The Machine Code operations tell the computer to read from and write to memory locations, they can also tell the computer to carry out quite complicated maths. So for instance we could add two numbers held in separate memory locations and put the result in another memory location. Memory is very important to a Machine Code program. The actual screen you look at on an arcade game is the video hardware decoding an area of memory into a picture. Teletext uses numbers in memory to represent block graphics; by changing this memory with a Machine Code program we can animate the screen.

So now I can tell you what Assembler is.

Imagine writing a Machine Code program purely in numbers. Having to remember numbers representing operations such as add and subtract among hundreds of others is tough. In the olden days programmers used Machine Code handbooks (these are still given today when you buy an Assembler, the book is in a handy pocket form). They would look up the operation and write the associated number (Machine Code) into their programs memory area. What they were actually doing was hand ASSEMBLING. Nowadays Assembler programs allow you to type in the names of the operations, these names are abbreviations e.g. NO OPERATION is usually shortened to NOP (telling the computer to do nothing until the next time it fetches a number from memory). These abbreviations are called mnemonics. The Assembler does many other things

besides converting operations names into CODE which will be covered in a future issue.

Memory can be read and written to from within AMOS using Poke, Doke, Loke, Peek, Deek and Leek. Check out your AMOS manual for more information on these commands.

Lets write a program in AMOS to add two numbers together that does not use variables.

Reserve 1000 memory locations for bank 10

Reserve as work 10,1000

Set up the first memory location in bank 10 with the value 100

Poke start(10),100

Set up the second (first+1) with the value 50

Poke start(10)+1,50

Set the value of the third location to the sum of the contents of the first and second

Poke ➡ start(10)+2,peek(start(10))+ ➡ peek(start(10)+1)

Show the contents of the third location

Print peek(start(10)+2)

Peek and Poke only deal with one byte memory locations try Doke, Loke, Deek and Leek which deal with more.

If you are reading from and writing to memory always make sure you are writing to memory you have reserved. If you write to unreserved memory you might corrupt data that a program is using. If you do this the program using the memory won't work as it should, causing undesired results (like big red flashing boxes with strange names! Ed.).

Now as you have seen memory locations can store numbers which can represent all types of data. Actually the computer, represents these numbers internally in a different format from decimal. This format is known as binary.

Binary is based on the fundamental principal of computers. Computers basically act on switches which are true or false. When you turn on an ATARI ST (blasphemy! Ed.) you will notice that the switch has a 1 for the on position and a 0 for the off position marked on the switch (engineers denote 1 for true and 0 for false). This one switch indirectly triggers millions of switches and your programs alter switches in the computer thus changing its state. The invention and refinement of the transistor has meant that switching processes are now very reliable and consume negligible power and space.

Given eight switches how would you represent a number using these? If the switch is on then denote that with a 1 otherwise denote it with a 0. So if all switches are off we would write 00000000. If they were all on we would write 11111111. If only the right most switch were on then we would write 00000001. The following table shows one way we could represent numbers.

Decimal number Switches representation

Decimal number	Switches representation
0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
. . . .	
254	11111110
255	11111111

This should give you a feel for binary numbers, later on converting decimal to binary and other arithmetic to do with binary will be covered. You can check your results in AMOS using the Bin$ function which is documented in your AMOS manual. I'll show you an easy way to work out binary numbers in the next issue.

Each memory location contains 8 switches. Actually 8 capacitors (components for storing electric charge) are usually used. When a capacitor is charged this represents 1, when not 0. Capacitors slowly lose charge so have to be topped up. RAM, Random Access Memory has millions of these capacitors and this top up is called RAM refresh. The computer makes this refresh process part of its system but this is a pretty antiquated way of refreshing memory. Nowadays RAM is starting to take care of this refresh process itself.

Each switch is a bit (BInary digiT, a number which can have two values). Eight bits make a byte, 2 bytes (16 bits) make a word and 2 words (4 bytes (32 bits)) make a longword.

By looking at two memory locations together we can have a (2 times 8 bits) 16 bit number. This number can go over 255 and can get quite large. By looking at more bytes together we can get an even larger range of numbers; in fact the whole of memory can be looked at to contain one number.

All the amazing things you see on a computer are attributed to the bit. A compact disc player takes 16 bits off a disc, treats it as a number representing a proportional voltage sent to a speaker which in turn pushes air. These 16 bits are read at a high rate every second called the sampling frequency resulting in a rapid number of pushes on the air

per second. Rapid pushes in air rapidly push your eardrum in a similar fashion resulting in your brain indirectly decoding the compact disc into music. The reason for me telling you this is to show that games can be quite boring in comparison to the amazing practical things computers can be used for. Also this is only the beginning, in the future computers will be an extension of the brain. In the above example a compact disc player could bypass the eardrum straight into the brain. Besides entertainment, blindness and deafness could be 'cured' and many previously undreamed of creations created. So dump the joystick (unless its for testing your program) and realise your dreams through computer technology.

If you have any comments on this column (good and/or bad) or interesting example programs please write in.

For these articles we will be using the Amos Assembler which is available from Deja Vu Software, it will allow you to write Assembler within AMOS and is more than adequate for most Assemblers. The version from the library contains more examples than found on the version with the compiler. On the other hand for a full professional Assembler (if you want to write 100% Assembler) it has to be Devpac 2 from HiSoft; although not very fast it doesn't have UNFORGIVABLE bugs compared to some other fast professional Assemblers (it also has a very good debug program).

SHOPPING LIST

For people who really want to learn more than we can cover in "ALL ABOUT AMOS" there are some books available which you may find useful.

'AMIGA Hardware Reference Manual'
Published by Addison Wesley.
ISBN 0-201-18157-6

'AMIGA ROM KERNEL Reference Manual: INCLUDES & AUTODOCS'
Published by Addison Wesley.
ISBN 0-201-18177-0

'AMIGA ROM KERNAL Reference Manual: LIBRARIES & DEVICES'
Published by Addison Wesley.
ISBN 0-201-18187-8

'Mapping The AMIGA'
Published by Compute! books.
ISBN 0-87455-128-5

The ABACUS range of books for the AMIGA are also very good with plenty of examples in Assembler.

WALLET WARNING!

These books will seriously deplete your bank balance.

QUESTIONS?

If you want to ask Gary a question about Assembler or the AMOS internals feel free to drop him a line here and we'll see that it gets to him!

Unfortunately due to working committments Gary will only be able to answer queries through this magazine.

THE SOUND OF MUSIC...

BY ANNE TUCKER

There has been a large gap in the range of AMOS educational software - there hasn't been anything to teach children about music.

Now, at last, that gap has been filled by 'Music Box'. Not every kid has the chance to learn an instrument at school, so they cannot find out the fun that can be had by playing about with notes and instruments, either trying to play a favourite tune, or trying to make a new one.

The program has been set out with a great deal of thought, you don't need to know ANYTHING about music before you load Music Box, but you should have learnt quite a bit before you finish with it.

On loading, you are given three options:-

A Bit About Music
Help with Music Box
Music Box Program

If you are a complete musical beginner, then choose A Bit About Music. Here you are given a tutorial on the basics of music, things like note lengths, terms used in music, and everything else you will need to get you on your way. The layout of this section makes for easy reading, the font is clear and the lines well spaced, there's no need to screw your eyes up to read the screens, which makes it great for children to follow. There are simple diagrams to illustrate the text.

The Help section is set out in the same clear style as the one above and takes you through the many functions of Music Box. Each aspect is clearly explained with nothing being left to the imagination.

After loading the main program, you see a screen which is divided into two sections. The top displays the musical score, the bottom displays the control panel. The controls are displayed as clickable buttons, to access each function.

To start putting music onto the page, click on the note length you want from the bottom row of buttons, and then click again to place it onto the page. Clicking the right mouse button will take the current note off your pointer ready for you to choose a new option. Keep a count of the note lengths you place as you go along, as you will not be allowed to put too many in each bar.

You can test the sound of each note before you place it by pressing 'T' on the keyboard. If you do make a mistake, you can 'kill' it (a case of tone death? Ed.) by clicking on the delete button, lining up the gunsight with the offending note and shooting it!

The first button is used to select your instruments. Clicking here gives you a new control panel where you can mix your choice of four instruments out of the wide choice available to you. To test your mastermix, press the

15
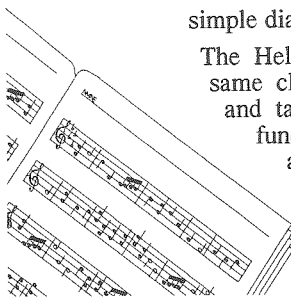
spacebar and the tune currently displayed will be played to you, tapping the spacebar again will stop the music.

To alter the speed at which the music is played you summon the speed control panel where you can increase or decrease the playing speed and test the result before going back to the main panel.

If you'd like to play music on the keyboard, then click on the button and play away! There is a choice of two time signatures, 4/4 or 3/4. Clicking a button changes from one to the other.

You can play your creation or stop it using the cassette player style buttons, get help with the 'H' button or throw everything away with the bin button. Saving and loading tunes is accomplished easily and there are several tunes on disk for you to listen to.

A really handy learning aid is the option to have the notes named by letter as you enter them onto the page. This will help you to remember what each line and space represents as you use the program.

All combinations of notes are covered along with their corresponding rests.

After using this program, I can't think of anything it lacks! Everything is so well set out that it's impossible to fault. As far as I can see, anyone wishing to learn the rudiments of music will benefit greatly from using this program. It's easy to use so will quickly give results to even the youngest user.

This program is a must for those wanting to get the feel of music before progressing to the more complicated music creators available (I assume that's a strong recommendation, Anne? Ed.).

RUNNING RUMOURS

SOUNDTRACKER WITHOUT THE HASSLE!!

A whisper on the grapevine tells me that a new AMOS extension has been written which allows the user to play Soundtracker music modules without first converting them into the AMOS music bank format.

This of course means that users will not have to go through the process of saving their files from Soundtracker, loading the AMOS conversion program and then waiting ages for them to be be processed (and even then some still would not play properly!).

Details are sketchy but it looks as though it may become available just after Christmas, keep an eye on these pages over the next couple of issues.....

THE 3D ANGLE

An on-going comprehensive guide to 3D graphics generation using AMOS

BY COLIN WHITE

Have you ever dreamt of producing three dimensional graphics to rival those of Starglider, Xiphos or Elite quickly and simply on your Amiga? If so, your dreams can now become reality. AMOS 3D enables the realisation of such 3D vector graphics without the hassle of using machine code or complex mathematical algorithms.

SIMPLE SCREEN COORDINATES

Let us start with a simple object (represented by an asterisk (*) in the diagrams). In a two-dimensional "world" (such as those you may already be used to programming platform games with, for example) the location of an object is defined according to its position on a 2D graph (see Figure 1). Any point can be defined starting from (0,0) (known as the ORIGIN); moving along the X axis for the horizontal position and then along the Y axis for the vertical position. Most screen coordinates originate from the top left-hand corner of the screen. In Figure 1, the asterisk is 10 to the right of the origin and 10 below the origin,

Figure 1

Figure 2

therefore its location is written as (10,10), where the first number represents the position on the X axis and the second refers to the Y axis position. Objects can be moved up and down (incremented or decremented on the Y axis) or left and right (decremented or incremented on the X axis). Therefore, to move the asterisk diagonally down to the left we could subtract 4 from the X axis and add 4 to the Y axis. This would take the asterisk to point B on the graph.
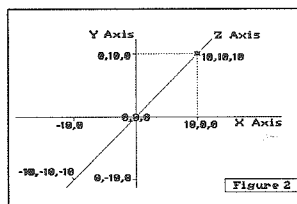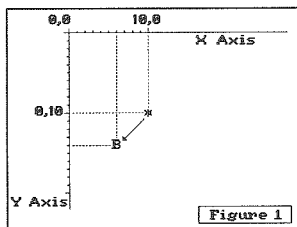
AND NOW....3D!

With 3D, however, a different screen coordinate system is used and a new plane of movement is added. Figure 2 illustrates the 3D screen coordinates system, with the origin being in the centre of the screen. This enables all three axes to have either negative or positive values. The Z axis allows objects to be moved closer or further away from the foreground of the screen, adding the so-called Third Dimension of depth which gives 3D its name. Objects located in a 3D "world" will have their locations represented in the form (X,Y,Z) with the value of Z being the distance of the object from the

17

foreground. To illustrate this we return to the example of the asterisk. On the 3D graph its location is given as (10,10,10). To move the object closer to the foreground the value of Z is decremented, while incrementing the value of Z causes the asterisk to be moved further away.
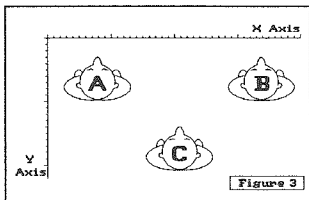
Figure 3

EVERYTHING IS RELATIVE....

Of course, all of the above coordinates assume that all movements are taken from a fixed point (0,0), but in the real world objects are moving around each other all the time. A true 3D system must reflect this. Figure 3 illustrates the concept of relative coordinates. From the viewpoint of Person A; Person C is behind and to the right (+x,-y). However, from the position of Person B, Person C is behind and to the left (-x,-y). Similarly, in AMOS 3D, you can specify a given point from which to view an object and this becomes the new origin from which all the X, Y and Z axis points originate.

Are you following this so far? If so, good, but if not, in the words of the immortal Douglas Adams - ''Don't Panic!'' - 3D object programming can take quite some getting used to but it needn't be considered too daunting. Just remember to study the AMOS 3D manual, persevere and above all, experiment! And don't forget to read the 3D columns that appear in this magazine as well, of course....

SPLAT!!!

Don't you just love all of those wonderful screens rolling on in a myriad of ways from out of nowhere? I have made a collection of about two dozen different ways of making screens appear, and just for you here is the first!! It splerges the screen line by line from the top of the screen to the bottom. Just type it in and have fun!

```
Screen Open 1,320,200,2,Lowres
Curs Off
Cls 0
Palette $0,$FFF
For L=1 To 20
Centre At(,L)+"ALL ABOUT  ➡
AMOS SPLERGE ROUTINE"
Next L
Screen Open 2,320,200,2,Lowres
Curs Off
Cls 0
Palette $0,$FFF
SPLERGE[2,1,2]
End

Procedure SPLERGE[SPEED  ➡
,SOURCE,DEST]
Screen SOURCE
SOURCE_SIZE=Screen Height
Screen DEST
DEST_SIZE=Screen Height
V=Min(SOURCE_SIZE,DEST_SIZE)
Screen SOURCE
SOURCE_SIZE=Screen Width
Screen DEST
DEST_SIZE=Screen Width
H=Min(SOURCE_SIZE,DEST_SIZE)
For LOP=V-SPEED To 0 Step-  ➡
SPEED
For LOP1=0 To LOP Step SPEED
Screen Copy SOURCE,0,LOP  ➡
,H,LOP+SPEED To DEST,0,LOP1
Next LOP1
Next LOP
End Proc
```
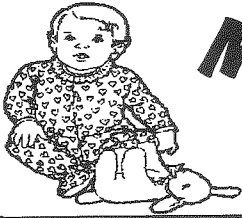
MUMMY, WHAT IS A COMPUTER?

Okay, so you've just bought and installed your first computer, and as you move aside the miles of bubblewrap and excess packaging (obviously the green revolution has yet to hit the computer industry) you wonder: What exactly is a computer? What can and can't it do? In the broadest sense, a computer could be described as an electronic machine which processes data, but this can also apply to other devices, for instance calculators. To be more precise in our definition we will need to examine all the components of a computer system, and the tasks which they are designed to undertake, so over the next few issues, we'll be leading you through the complex but exciting world of computing starting with an examination of the computer itself, and moving on to look at all the applications of computing in todays world. Scattered throughout this magazine you will also find information boxes explaining the origins and definitions of computer related terms and phrases. In future columns we'll also be getting to know the Amy much better with a look at the Amiga hardware, Command Line Interface (CLI), Workbench and peripherals.

HARD, SOFT OR JUST FIRM?

The solid physical components of your computer system are known as HARDWARE (see Figure 1). Some of these are essential for your Amiga to operate, while others are optional and enable you to extend the use of your computer for specific purposes. The programs that you use with your computer are known as SOFTWARE. As with hardware, some items of software are more important to the functioning of your computer than others. Programs known as SYSTEM software or UTILITY programs are needed to tell the computer how it should operate (how it communicates with other devices such as printers for example). Other types of programs are used by you the eager user to create documents with word processing and text editing software; pictures with graphics packages; and even your own pieces of software using programming languages (such as the Amiga's very own AMOS). For those of you who are real jargon junkies, there is also a third category, that of FIRMWARE. These are programs which are permanently installed on microchips within the hardware itself. Examples of firmware in operation include the good ol' hand an' disk Workbench prompt which loads as soon as you start up your Amiga.

THE HARD FACTS:

INPUT AND OUTPUT DEVICES: Input devices are used to give commands or information to the computer, while output devices are used to receive data back from the computer. Some devices are used for both input and output, such as touchscreens. Touchscreens are a specially built type of VISUAL DISPLAY UNIT (see below) which can respond to the fingertip touch of the user and display the information as he or she points to it.

TYPES OF HARDWARE: Here is a

summary of the most common types of hardware. Future columns will examine each item of hardware in more depth.
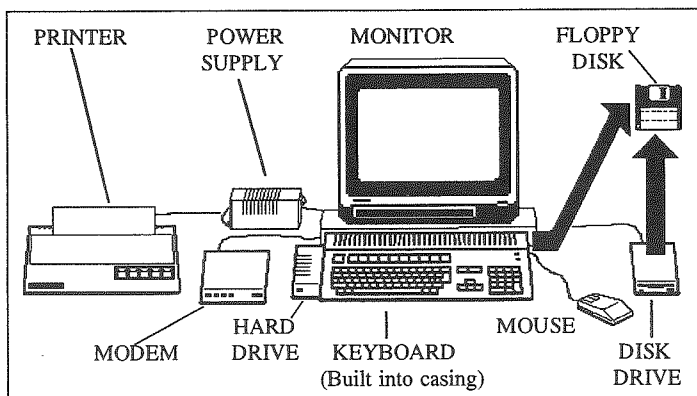
V.D.U. (Visual Display Unit) or monitor screen. The Amiga can also be connected to standard television sets. The VDU displays information on the screen in pictures made up of tiny dots known as PIXELS (PIcture ELements). The smaller these pixels are, the more detailed the picture is. Ordinary VDUs and televisions are, of course, output devices only, but new technology has lead to the increased development and use of touch sensitive screens which enable both input and output of information.

MODEM. Short for MOulator-DEModulator, a modem can convert the signal from your Amiga to a telephone signal and back again. This enables data to be sent from one computer to another across a telephone line.

PRINTER POWER SUPPLY MONITOR FLOPPY DISK

MODEM HARD DRIVE KEYBOARD (Built into casing) MOUSE DISK DRIVE

KEYBOARD. Input is received from the keys which the user presses, and this information is passed to a device which translates the commands given by the keyboard operator into a form which the computer can process.

MOUSE. A hand held device for inputting information. Using the mouse you can move an object called a POINTER (usually represented as a small arrow) around the screen to select items.

DISK DRIVE. The disk drive is a device which can write data to a disk and read it back again. An INTERNAL disk drive is built into the same casing as the computer itself, while an EXTERNAL disk drive is a self-contained unit which is connected to the computer via a cable.

PRINTER. Printers are used to get permanent displays of information from the computer on material such as paper, card or fabric. The appearance of these displays (known as PRINTOUTS or HARD COPIES) differs according to the type of printer used to produce them. Future issues will provide fuller information about printers, which really require more detailed coverage than can be given in this issue's column.

STORAGE DEVICES: Storage devices are used to store information during or after processing by the computer. The amount of data a storage device can hold and the speed at which it can store or retrieve the data varies considerably from one device to another.

FLOPPY DISK The standard form of portable data storage is the floppy disk,

a small, flat, plastic or card slip containing a disk of silicon. This silicon disk is magnetized and is the part of the disk that holds all the data.

HARD DISK (or Winchester disk) The hard disk is a solid magnetic disk capable of holding much larger amounts of data and handling them at a faster rate than a floppy disk, but is not as easily portable. A hard disk can be either INTERNAL or EXTERNAL, that is it can either be installed in the same casing as the computer or it can be connected to the computer via an expansion port.

That's a brief look at the major pieces of hardware available, but don't worry if you still feel in need of a greater explanation, we'll be spending several issues covering all the above items of hardware in fuller detail. Next ish we'll be looking at the various types of software available and how they interact with each other and with the hardware they are used on.

If you would like us to examine and explain any aspect of computing in greater depth, just drop us a line.

MY DEFINITION

BASIC

BASIC is an acronym for Beginners All purpose Symbolic Instruction Code. The very first version of the BASIC language was written by T. E. Kurtz and J. G. Kemeny in 1964. It was initially developed as a language to teach programming and was derived from the scientific language FORTRAN (which is another story all-together!). You don't want to know this but the only home micro computer that didn't come with a version of BASIC as a standard language was the ill-fated JUPITER ACE which came with a version of the FORTH language (and trust me when I say that you never want to program in that!).

MORE ABOUT AMOS

We are not the only source of AMOS information, there is also the OFFICIAL AMOS CLUB (which offers a phone based helpline) and a disk based magazine called "TOTALLY AMOS".

Now you may think it is strange that I would give a free plug to the 'competition' so to speak, but these two publications offer some excellent services (which you may find useful) and they have both helped me get started with "ALL ABOUT AMOS".

The **AMOS CLUB** can be contacted at

**1 LOWER MOOR
WHIDDON VALLEY
BARNSTAPLE
NORTH DEVON
EX32 8NW**

"TOTALLY AMOS" can be contacted at
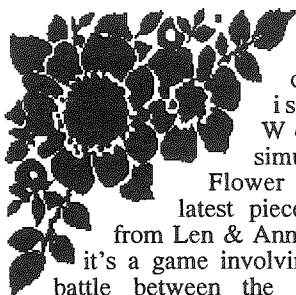
**1 PENMYNYDD ROAD
PENLAN
SWANSEA
WALES
SA5 7EH**

WHERE ARE WE?

We can be found busily typing away at:

36 CLEVERLY ESTATE WORMHOLT ROAD LONDON W12 0LX ENGLAND

THE BIRDS & BEES

BY LEO DOUGLAS

Hey, man, far out - oops, no this isn't a Woodstock simulation - it's Flower Power, the latest piece of software from Len & Anne Tucker, and it's a game involving the eternal battle between the gardener and various beasties determined to trash his lovely blooms!

POTTY GARDENER?

You are a mild-mannered gardener who's pottered out into your back garden on a lovely sunny day to plant some seeds and raise a few pretty flowers for a forthcoming flower show. Unfortunately, there seem to be a number of rather nasty mutant bees, ladybirds and bulldogs (yes, bulldogs!) who are intent on ruining your prize plants so you'll have to quickly water them until they grow to maximum height then dig 'em out and pop them safely into a basket. Sound simple enough? Don't be so sure. The flowers need more water than yer average inferno (sort of) and the basket and water tap are at opposite ends of the garden! So off you go tending your blooms and swatting the pests as they try to expunge your little leafy babies.... There are ten levels to the game (your fault for entering ten different flower shows, you "sucker"!).

FLOWER FIGHT!!

Once you get into the game it's quite easy to get engrossed in encouraging your little flowers to reach their best and whizzing about fighting off the pests that threaten them, but it can be tedious not to say infuriating planting a seed and watering it carefully only to have a rampaging bug or beasty come along and kill it off. (Incidentally, the nasty creatures not only devour your plant but they somehow manage to replace the turf where your seed was planted, leaving absolutely no trace of your horticultural effort!)

Help is provided from the text file on loading the game, and also with a command summary given as one of the options on the loading screen.

The Flower Power graphics are simple and cartoony. The animations on the beasties and on your own man are amusing if slightly too fast, but this is a graphically colourful and cheerful game.

GROOVY TOON...

An upbeat little toon from an APD disk is used as the background music - very jolly if slightly over-generous on the bass drum - and an intriguing(!) sampled voice saying "Go For It!" bursts into your ears whenever you have successfully grown a nice healthy specimen of planthood and are about to carry it off to the safety of your flower basket. When swatting the pests your versatile gardener gives a cry of "Ah so!", demonstrating his black belt karate chop picked up after a few months at the Gardeners Self Defence evening classes, no doubt. Not surprisingly, the battered beasties give an injured cry when successfully hit,

before whizzing off the screen. Various other samples which add further interest include a spade clanging and digging sound effects, all of which contribute to the enjoyment and realism of the game.

Although generally an excellent game the screen fades are a bit jerky, fortunately the screen scrolling at the end of the game is very smooth and professional.

JUST FOR FUN?

The intro text file proclaims Flower Power to be "Number One in a Just For Fun! series", and it is certainly a good bit of software-fun if slightly weird. That poor old gardener must end up getting totally knackered what with all the karate chops and running up and down his garden path chasing bugs AND carrying freshly picked flowers all over the place!. Well worth a look. One final word to Len & Anne - I'm afraid bees and ladybirds aren't technically garden pests - they don't eat the plants, they help 'em! Never mind....

INFO BOX

TITLE: FLOWER POWER
by SPARX (LPD 27)

REQUIRES: 1MB.
Joystick only.

available from:
Déjà Vu Software
25 PARK ROAD
WIGAN
WN6 7AA

More information can be obtained by contacting Sandra Sharkey on:

☎ 0942 495261

SCREECH!!!

Hands up all of those people who have had trouble with the Sample Bank Maker which came with AMOS, emmm.... Quite a few of you.... Ok well, the only real solution at the moment (at least until the Sample Bank Maker gets rewritten) is to load your samples as raw data. Fortunately AMOS can handle this quite well, the following program will pop up a little file selector for you to choose a sample. Once you have clicked on the one you want the procedure _SAM[] will load and play it. Try changing the value passed into the variable SPEED to alter the pitch of the sample and also to create some weird effects. If you own a sound sampler (or just the software) like Master Sound, Techno Sound or Audio Master why not build your own sample banks by cutting and pasting different samples together? You can then use the SAM RAW command to play each individual sample.

```
F$=Fsel$("","","")
_SAM[F$,8000]
End
Procedure _SAM[N$,SPEED]
Open In 1,N$
L=Lof(1)
Close 1
Reserve As Chip Work 16,L
Bload N$,Start(16)
Sam Raw 15,Start(16),L,SPEED
Erase 16
End Proc
```

SPACE, THE FINAL MATHS TEST.....

BY COLIN WHITE

ROCKET MATHS is an Educational maths program, containing levels for addition, division, multiplication and subtraction, and including a choice of three levels of difficulty providing suitability for children of almost any age group.

TIME FOR LAUNCH?

The player must guide an intrepid astronaut through the launch of his spacecraft (Level One), across a galaxy of planets (Level Two) and an asteroid storm (Level Three) and safely down onto a landing pad (Level Four) by correctly answering the sums that the computer displays on a small monitor in the bottom left-hand corner of the screen.

A high score table provides an additional element of competition for the player to compete against their friends and/or their own previous best scores.

I NEED HELP....

On screen instructions are given at the start of the game. There's no access to the instructions during the game, but by the time you're well into it the controls are straightforward.

A couple of problems do exist with the control method. Number selection when answering sums relies on left/right movement of the joystick and could prove over-sensitive for children. And there's no option for correcting a wrongly selected digit. Tut tut....

PICTURE THIS....

Graphics are well presented, with some interesting reward screens and animations to keep the player watching. The loading, intro and "game over" screens and also the monitor graphics are very professional in appearance and would not be out of place as part of a full-blown commercial game, making use of those kind of shades of grey and green that are often used for hi-tech, futuristic space or military type graphics. These contrast very well with the more cartoony graphics used for the astronaut and the lunar landscape.

BELOVED SOUNDS

The astronaut gives a cheerful "Hello" and waves before boarding his rocket, (The "Hello" sounds remarkably like a snatch from a song of the same name by a group called The Beloved a coupla years back - yes, I have got an encyclopaedic knowledge of pop music!) and appears to dematerialise from the launch pad with a noise reminiscent of a Star Trek transporter beam or similar, only to reappear inside the rocket. A whooshing sample has been used for both the lift-off and for the planet-hopping of the second level, and while it works well on the latter, it doesn't give a very convincing feel on the lift-off sequence. One other minor criticism - the sampled sound used in response to the user's input is the same

whether he or she gives a correct answer to the sums or not. Now you can call me pedantic ("Hello, Pedantic" - A.A.A. Readers) but in my opinion this is a little misleading for kiddies since they rely on sound clues as well as visual clues to reinforce the validity of their answer.

The flow of the game is not too fast for younger or slower users to cope with but not too slow for those of us not used to waiting about (like impatient software reviewers who've got piles of disks to go through!). The brief waits during periods of disk accessing between sequences can easily be coped with and screen fades are smooth and swift.

THE FINAL COUNTDOWN
A well written and presented little piece of software - if this program is anything to go by, more educational software from this author would be most welcome.

BOING!!!

AMAL is probably the most advanced feature of AMOS. Just to prove the real versatility of this sub-language check out this mini program which was written by François Lionet. All you do is select a screen to load and then go to direct mode, then AMAL will take over and bounce the screen everywhere!!!

```
Channel 0 To Screen Display 0
Channel 1 To Screen Offset 0
F$=Fsel$("","","Pick a picture")
Load Iff F$,0
A$=A$+"Debut:Let Y=-256"
A$=A$+" Let R0=256"
A$=A$+" Let R1=8"
A$=A$+" Let R2=45"
A$=A$+" Move 0,R2-Y,R1"
A$=A$+"Loop:Move 0,R2-Y-R0,R1"
A$=A$+" Move 0,R2-Y,R1"
A$=A$+" Let R0=R0/2"
A$=A$+" Let R1=R1-1"
A$=A$+" If R0 Jump Loop"
A$=A$+" For R0=0 To 25"
A$=A$+" Pause"
A$=A$+" Next R0"
A$=A$+" Move 0,320,50"
A$=A$+" Let RA=RA+1"
A$=A$+" Jump Debut"
B$=B$+"Debut:Let X=0"
B$=B$+" Let R3=RA"
B$=B$+" Let R0=320"
B$=B$+" Let R1=10"
B$=B$+" Move 0-X,0,R1"
B$=B$+"Loop: Move 0-X+R0,0,R1"
B$=B$+" Move 0-X,0,R1"
B$=B$+" Let R0=R0/2"
B$=B$+" Let R1=R1-1"
B$=B$+" If R0 Jump Loop"
B$=B$+"Sync:"
B$=B$+" If RA=R3 Jump Sync"
B$=B$+" Jump Debut"
Amal 0,A$
Amal 1,B$
Amal On
Direct
```

ALL THE FUN OF THE FAIR!

BY COLIN THE CLOWN

Roll Up! Roll Up! Come and see the amazing performing software!! This one's Big Top Fun, ladeez and gentlemen, a package of four educational programs set in a circus!!

Program 1: **WORD BALANCE**

A game to aid children who are learning to read. The central element of this program is a juggler on a unicycle, who carries a block with a word on it which the child must match to the correct picture from a selection of pics displayed on the screen. On level three, this is reversed and the juggler holds the picture, with a choice of words displayed for the player to choose from. An incorrect match causes the juggler to drop the block with the wrong answer. I did notice some repetition of pictures and words in the same game - 'cactus' appeared simultaneously in two adjoining tiles on level 3 - but on the plus side the juggler and unicycle animation is very amusing. Samples accompanying this program include a whooshing as the word/picture tiles slide onto the screen and an odd (if rather rude!) parping noise when the player selects a tile.

Program 2: **MATCHPLAY**

A variation on the good ol' pairs game - matching pairs of cards up to reveal a circus audience. A reasonable reworking of a classic game, with an interesting mix of word and picture cards on level 3. The card flips and fades when matches are made are smooth and speedy.
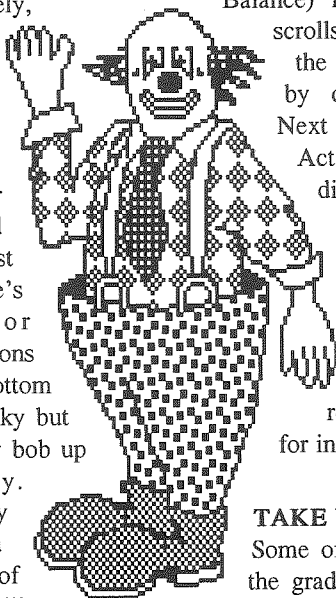
Program 3: **SEAL-A-GRAMS**

This game gives children guidance with their spelling. A clown throws balls with letters on them at a number of seals who the player controls, causing them to juggle the balls until the letters are in the right order spelling a recognisable word. The length of the words (and the number of seals) increases through the levels from 3 letters (3 seals) on level one to 6 letters (6 seals) on level three. The seals are attractively drawn and their animation is good if a little jerky in places when swapping letters. The control feature to swap the letters from seal to seal consists of clicking on the heads of the two seals you wish to swap between. Although this is pointed out in the introductory text on start-up, it is not indicated or emphasised anywhere within the Seal-A-Grams program itself. If the player is unable to mentally rearrange the letters and work out the word the program is looking for, a good hint facility is included. By clicking on an option marked ''Show me'' a picture of the

object is briefly dropped onto the screen as a visual clue.

Program 4: BALLOON BURST

Another word recognition program. The player controls a clown who must shoot at a row of lettered balloons which rise from the bottom of the screen to spell a word, bursting the balloons with the required letters. Unfortunately, the central character here, a gun-toting clown, is not too good. His animation doesn't flow very well at all, with his eyeballs swimming all over the place and his arms and legs animating in a most unnatural way! (Perhaps he's quadruple-jointed or something....) The balloons which rise up from the bottom of the screen are rather jerky but once they're in the air they bob up 'n' down very nicely. Shooting is accompanied by samples of a gun shot and a balloon pop when one of the balloons is hit. Controlling the clown could well be pretty tricky for kiddies. Problem is, the clown's got the proverbial wanderlust and is walking swiftly backwards and forwards across the screen so it takes some effort to master the timing for the mouse control, which involves waiting until the clown is underneath the balloon which you want to pop and pressing the left button. That isn't very easy especially after a few drinks (not that I have had a chance to try this!).

CIRCUS ATTRACTIONS

The method of program selection is a novel one. Instead of having a menu, users see a display board standing in the entrance to a big top. On the board, the first program in the quartet (Word Balance) is shown and the player scrolls through the four to select the program he or she wants by clicking on either "See Next Act" or "Choose This Act". There is a limited dictionary/set of pics used by Big Top Fun - the same words and pictures tend to crop up regularly throughout the programs, and sometimes several times in one game (the repetition in Word Balance, for instance).

TAKE YOUR PICS

Some of the pics just don't make the grade, but those that do more than make up for the few naff ones. The mini-pics, used on the tiles in the four programs as word clues are on the whole very attractive and most of the central characters (the seals in Seal-a-Grams, the juggler in Word Balance, etc.) are also adequately drawn and animated. The least engaging element of the graphics is the actual circus ring with audience. This same rushed-looking background appears in all the

programs and detracts slightly from the visual appeal of the main sprites in each program.

BIG TOP TUNES

The jolly circus/fairground music at the start of the programs, with applause and cheers on completion of a game or level, add to the feel of the game. The programs also contain a (thankfully) optional speech mode which speaks the answers in that now familiar(?) monotonous synthesised voice.

FLASHING POINTER!

The initial loading takes too long and there are some very dodgy fades/screen blackouts between programs. There are also a few dubious-looking requester boxes which open when the player completes a game. One more snag - the mouse pointer used by Big Top seems to be the default Amiga pointer. No problem there you may think, but the colour registers used by the pointer are also used for some murky shades of green (Ugh!) while on the loading screen one of the pointer colours is used for colour cycling on the big top lights so....voila! flashing mouse pointer ahoy! Oh dear....

FAIR'S FAIR....

Well, if you're not looking for technical excellence but you want a few cute, colourful and cheerful proggies for the kids with a fun educational slant, you could do worse than get a copy of Big Top.

MY DEFINITION

WYSIWYG

Pronounced "WIZIWIG" this weird computer word is really an acronym for What You See Is What You Get. It is used to describe the relationship between what is shown on the computer screen and what is output to a printer. WYSIWYG is an especially popular term in the field of Desk Top Publishing where the ability to layout your text and graphics are very precisely controlled. This magazine was produced using a WYSIWYG Desk Top Publishing package.